

Redmine 管理员手册

基于Redmine 0.9 版本

作者: 蒋鑫, 王胜
公司: 北京群英汇信息技术有限公司
联系: <http://www.ossxp.com/>
版本: 1.0
日期: 2010-04-19 17:10

目录

- 1 说明
- 2 项目管理
- 3 用户管理
- 4 用户组管理
- 5 角色和权限
- 6 问题跟踪系统
- 7 自定义属性
- 8 枚举值
- 9 配置
- 10 高级配置
 - 10.1 版本库配置
 - 10.2 发送邮件配置
 - 10.3 接收邮件
 - 10.3.1 方案
 - 10.3.2 各种方案的详细说明
 - 10.3.3 工作原理
 - 10.3.4 例子演示
 - 10.4 发送提醒邮件
 - 10.5 LDAP 认证
 - 10.6 配置单点登录
 - 10.7 从LDAP同步账号
 - 10.8 Subversion 基于路径授权

1 说明

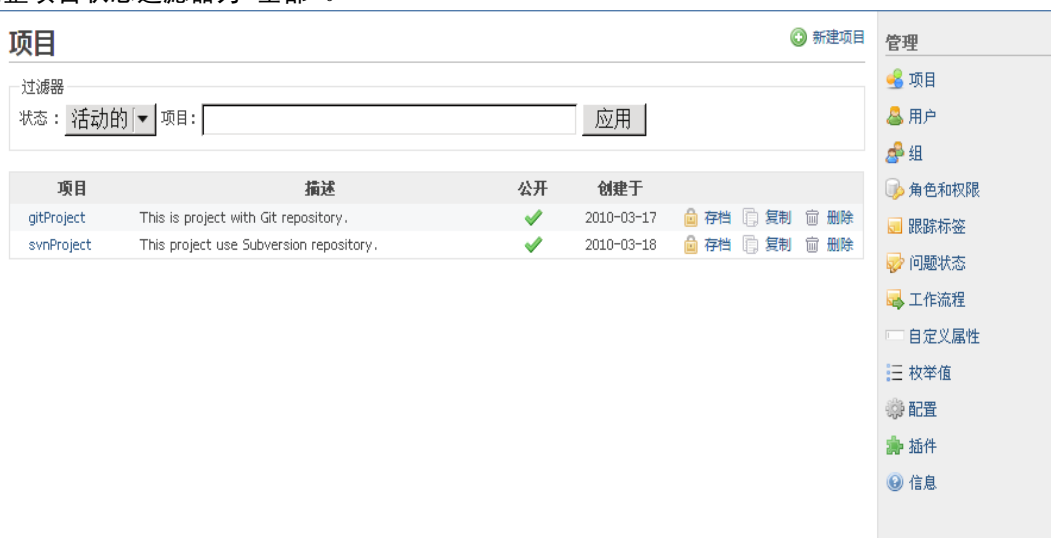
关于手册中方框标记部分的说明

- * Note 是需要注意的部分。
- * Hint 是群英汇为Redmine扩展的部分。

2 项目管理

* 项目列表

默认情况下，项目列表显示所有活动的项目。要想浏览所有项目(包括活动和归档项目)，需要调整项目状态过滤器为“全部”。



项目列表截图

- * 新建/编辑项目
- * 项目存档

在项目列表页面点击“存档”链接，可将该项目存档。存档项目对用户不可见。管理员可以取消存档。当父项目存档，它的所有子项目也都自动存档。

* 复制现有项目

在项目列表页面，点击“复制”链接复制你想要复制的项目。

复制项目的表单将会预填原项目的设置(如:跟踪标签，自定义属性...)。在该表单的底部，你可以选择复制原项目的哪些内容。

* 删除项目

在项目列表页面，点击“删除”链接，界面将跳到删除页面，选中“是”复选框，然后点击“删除”按钮，该项目以及子项目的所有数据都将被删除。

Note

存档与删除的区别：

- * 存档只是隐藏项目及子项目，而项目的数据还存在。
- * 删除则是将项目以及子项目的数据一同删除，无法恢复。

3 用户管理

* 用户列表

用户

过滤器

状态: 活动的 (4) 用户: 应用

登录名	名字	姓氏	邮件地址	管理员	创建于	最后登录
admin	firstname	lastname	wangsheng@ossxp.com	✓	2010年3月17日 星期三 10:30:30	2010年4月08日 星期四 17:27:23
hello	hello	hello	hello@bj.ossxp.com		2010年4月02日 星期五 15:55:36	2010年4月06日 星期二 15:49:33
jiangxiKin	jiang	jiang	wangsheng@ossxp.com		2010年4月02日 星期五 15:47:04	2010年4月06日 星期二 18:18:27
wangsheng	王	wangsheng	wangsheng@bj.ossxp.com	✓	2010年3月17日 星期三 14:46:37	2010年4月06日 星期二 18:41:53

(1-4/4) | 每页显示: 25, 50, 100

管理

- 项目
- 用户
- 组
- 角色和权限
- 跟踪标签
- 问题状态
- 工作流程
- 自定义属性
- 枚举值
- 配置
- 插件
- 信息

用户列表截图

默认情况下显示 活动的 用户，你可以调节用户状态过滤器，选择浏览不同状态的用户。通过点击右侧“锁定”链接，可将用户从 活动的 状态变为 已锁定 状态。同样点击“解锁”链接，可以将用户从 已锁定 状态变为 活动的 状态。

* 添加用户

点击右上角的“新建用户”链接，跳转到新建用户页面：

必填字段：

- * 登录名
- * 名字
- * 姓氏
- * 邮件地址
- * 管理员

选中之后，该用户将是管理员的身份。

Hint

从LDAP同步账号

Redmine的LDAP认证模式是：首先管理员配置好 [LDAP认证](#)，并且选中 即时用户生成 选项，然后当用户用LDAP账号第一次登录Redmine时，系统自动将LDAP里的用户信息存入数据库，为用户创建Redmine账号。

但这样会有一个问题：如果管理员想批量给某一项目添加项目成员，这时不得不一一通知这些人，让他们都立马登录一下Redmine（因为只有这样，他们才拥有Redmine账号，管理员才能给这些用户划分角色，授予权限）。

显然这很麻烦，给管理人员带来不必要的烦恼，似乎管理人员的权利需要员工给予。

群英汇给Redmine的管理人员带来了福音，我们添加了管理人员可以直接从LDAP服务器同步账号到Redmine的功能。

操作步骤，参见 [从LDAP同步账号](#)

* 认证

如果你已经配置了LDAP，那么认证区域将会出现一个“认证模式”的下拉选择框。Redmine 提供了两种认证机制

- * 内部认证

将用户的密码经过加密后存入数据库中。

- * 外部认证(如LDAP)

用户的密码存储在一些外部认证中，如LDAP。

- * 给用户发送账号信息

选中它，创建账号后，新建的用户将收到一封关于账号信息的邮件。

- * 编辑用户

想要编辑用户信息，只需点击用户名，就跳转到修改该用户信息的页面了。

- * 一般

属性和新建用户界面的属性一样。

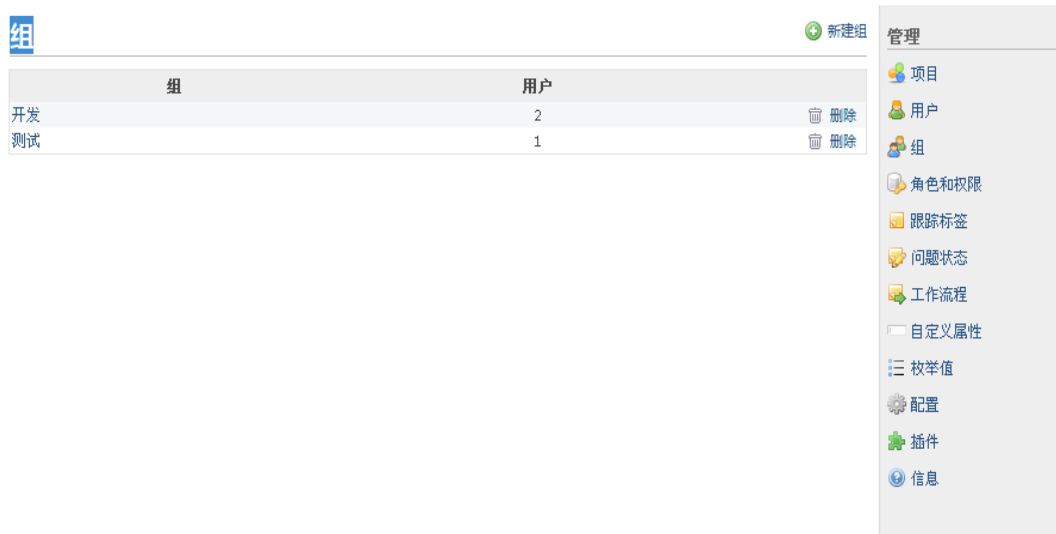
- * 组

如果你创建了组，在这里会显示组选项卡。

- * 项目

在这里可以给该用户指定项目。

4 用户组管理



用户组列表截图

从Redmine-0.9 开始，管理员可以定义用户组。

用户组可以像添加常规用户一样添加到项目中去。

当把用户添加到一个已存在的用户组里，该用户将拥有该组的一切权限。

5 角色和权限

角色决定用户在项目中拥有的权限。每个项目成员在项目中可以有一个或者多个角色。每个用户在不同的项目中可以有不同的角色。

管理员可以创建新的角色；也可以编辑已经存在的角色；只有当某一角色下没有用户时，该角色才能被删除。

* 角色的属性

* 名称

* 问题可指派给此角色

如果不选，问题就不能指派给该角色里的成员。

* 从以下选项复制工作流程

新建的角色可以从已经存在的角色中复制 workflow，如果不选则新建角色的 workflow 为空。

* 权限

当编辑一个角色的时候，你可以通过(选中/取消)那些不同的行为给该角色定义权限。

权限	描述
项目权限	
新建项目	允许非管理用户创建根项目
编辑项目	允许用户编辑项目的属性
选择项目模块	允许用户开启/禁用项目的模块

管理成员	允许用户添加/删除项目组成员或者改变项目组成员的角色
管理版本	允许用户添加/删除项目的版本
新建子项目	允许用户为该项目创建子项目
讨论区权限	
管理讨论区	允许用户添加/编辑/删除项目的讨论区
发表帖子	允许用户在讨论区发表帖子
编辑帖子	允许用户编辑任何帖子
编辑自己的帖子	仅允许用户编辑自己的帖子
删除帖子	允许用户删除任何帖子和回复
删除自己的帖子	仅允许用户删除自己的帖子和回复
文档权限	
管理文档	允许用户添加/编辑/删除项目的文档
查看文档	允许用户查看文档
文件权限	
管理文件	允许用户添加/编辑/删除项目的文件
查看文件	允许用户查看文件
问题跟踪权限	
管理问题类别	允许用户添加/编辑/删除问题类别
查看问题	允许用户查看问题
新建问题	允许用户新建问题
更新问题	允许用户编辑问题
管理问题关联	允许用户添加/删除问题之间的关联
添加说明	允许用户给已存在的问题添加说明
编辑说明	允许用户编辑任何的问题说明
编辑自己的说明	仅允许用户编辑自己添加的问题说明
移动问题	允许用户将已存在的问题移动到别的项目中
删除问题	允许用户删除问题
管理公开的查询	允许用户添加/编辑/删除公开的问题查询
保存查询	允许用户保存自定义问题查询
查看甘特图	允许用户浏览甘特图
查看日历	允许用户浏览日历
查看跟踪者列表	允许用户查看问题的跟踪者
添加跟踪者	允许用户给已存在的问题添加跟踪者
删除跟踪者	允许用户删除问题的跟踪者
新闻权限	
管理新闻	允许用户添加/编辑/删除项目的新闻
为新闻添加评论	允许用户给新闻添加评论

版本库权限	
管理版本库	允许用户配置项目的版本库
浏览版本库	允许用户浏览并查看项目版本库的内容
查看变更	允许用户查看变更集
访问提交信息	给予Redmine.pm具有访问项目版本库写的权限
时间跟踪权限	
登记工时	允许用户给项目登记工时
查看耗时	允许用户查看项目的耗时历史
编辑耗时	允许用户编辑任何耗时记录
编辑自己的耗时	仅允许用户编辑自己的耗时记录
管理项目活动	允许用户添加/编辑/删除活动
Wiki权限	
管理Wiki	允许用户创建/删除项目的Wiki 删除Wiki页面是不可恢复 的操作，它会删除所有相关的页面及附件
重命名Wiki页面	允许用户重命名Wiki页面，或者指定Wiki页面的父页面
删除Wiki页面	允许用户删除Wiki页面，此操作是不可恢复 的，页面 以及修改历史将被统统删除
查看Wiki	允许用户查看Wiki页面
查看Wiki历史记录	允许用户老版本的Wiki页面，或者查看不同版本之间的差别
编辑Wiki页面	允许用户编辑没有保护的Wiki页面
删除附件	允许用户删除Wiki页面的附件
保护Wiki页面	允许用户锁定/解锁Wiki页面，并具有编辑已锁定的Wiki页 面的权限

* 系统角色

在Redmine共有2种系统角色，他们供Redmine内部使用，因此不能被删除。

* Non member

非项目成员角色，用于定义那些不属于该项目成员的用户权限。

* Anonymous

匿名角色，用于定义匿名用户在项目中的权限。

Note

* 这两个角色仅被用于公开项目，因此匿名用户和非私有项目成员用户不能看到它。

* 一些权限不能赋给这两个角色。例如“管理成员”

6 问题跟踪系统

* 跟踪标签

跟踪标签 » 新建跟踪标签

名称 *

在路线图中显示问题

从以下选项复制工作流程

项目

gitProject

svnProject

[全选](#) | [清除](#)

新建跟踪标签截图

跟踪标签代表不同的问题类型。默认情况下有三种标签：错误，功能和支持。管理员可以新建跟踪标签：

- * 名称
- * 在路线图中显示问题
- * 从以下选项复制工作流程

新建的跟踪标签可以从已经存在的跟踪标签中复制工作流。如果不选，那么新建的跟踪标签的工作流为空。

* 项目

可以指定新建的跟踪标签用于哪些项目中。

* 问题状态

管理员可以根据需要自由地添加和删除问题状态。

问题状态含有的属性：

- * 名称
- * 已关闭的问题

如果选中，则表示处于该状态的问题属于已经关闭的问题（可以声明多个状态属于已关闭的状态）。

* 默认值

指定新建问题的默认状态（只能有一个状态属于默认状态）。

* 工作流

工作流使你可以定义不同角色的项目成员对不同类型的问题所能做的问题状态变更。

工作流程

[编辑](#) [复制](#) [摘要](#)

选择角色和跟踪标签来编辑工作流程：

角色：[管理人员](#) 跟踪标签：[错误](#) 只显示被此跟踪标签使用的状态

[编辑](#)

当前状态	可变更的新状态					
	新建	进行中	已解决	反馈	已关闭	已拒绝
新建	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
进行中	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
已解决	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
反馈	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
已关闭	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
已拒绝	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[全选](#) | [清除](#)

[保存](#)

定制工作流截图

在工作流设置页面，选择你想要编辑的角色和跟踪标签，点击“编辑”按钮，跳转到编辑页面，此页面可以编辑该角色对该标签所能做的变更状态。左侧列出了该跟踪标签所有的状态，你可以在右侧选择针对该状态该角色可以变更的状态。

Note

为了让某一角色用户能够改变问题的状态，你需要给予他们 **可变更问题状态** 的权限。

7 自定义属性



自定义属性截图

Redmine允许管理员给以下实体添加属性，从而给这些实体添加额外的描述信息：* 问题 * 耗时 * 项目 * 版本库 * 用户 * 组 * 活动(时间跟踪) * 问题优先级 * 文档类别

自定义的属性有以下几种数据类型：

* 整数

* 字符串

单行输入框

* 文本

多行输入文本域

* 日期

* 布尔值

复选框

* 列表

下拉列表

自定义属性的约束条件：

* 最小-最大长度

决定自定义属性的字段长度(0表示没有限制)。

* 正则表达式

通过指定的正则表达式，对自定义属性的内容进行验证。

* 默认值

自定义属性的默认值

* 跟踪标签

设置该自定义属性可以在哪些跟踪标签中使用

* 必填

该自定义属性是否是必填字段

* 用于所有项目

该自定义属性是否用于所有项目中

* 作为过滤条件

该自定义属性是否作为过滤器的过滤条件

* 可用作搜索条件

该自定义属性是否可以用作搜索时的搜索条件

8 枚举值

枚举值					
活动 (时间跟踪)					
名称	默认值	活动			
设计		✓	▲ ▲ ▲ ▼ ▼		删除
开发		✓	▲ ▲ ▲ ▼ ▼		删除
测试		✓	▲ ▲ ▲ ▼ ▼		删除
新建枚举值					
问题优先级					
名称	默认值	活动			
低		✓	▲ ▲ ▲ ▼ ▼		删除
普通	✓	✓	▲ ▲ ▲ ▼ ▼		删除
高		✓	▲ ▲ ▲ ▼ ▼		删除
紧急		✓	▲ ▲ ▲ ▼ ▼		删除
立刻		✓	▲ ▲ ▲ ▼ ▼		删除
新建枚举值					
文档类别					
名称	默认值	活动			
用户文档		✓	▲ ▲ ▲ ▼ ▼		删除
技术文档		✓	▲ ▲ ▲ ▼ ▼		删除
新建枚举值					

枚举值截图

Redmine允许管理员对以下实体添加枚举值，从而扩充这些实体可选择的值：

- * 活动(时间跟踪)
- * 问题优先级
- * 文档类别

这些枚举值是跨项目的 管理员可以任意地添加/修改/删除。

枚举值包含的字段：

- * 名称
- * 活动

只有活动的枚举值才能被使用

- * 默认值

如果选中，那么在使用同类枚举值时该枚举值默认被选中

同一类枚举值的默认值只能设置一个

9 配置

- * 一般
- * 应用程序标题

显示在应用程序头部的文字

- * 欢迎文字

这些文本将显示在项目概述页面

- * 附件大小限制

上传文件的最大值。 默认为5120KB

- * 每页显示条目个数的设置

这个主要用于分页显示，在这里可以填入你想要每页显示的记录数，多个值之间用逗号分开。

默认值: 25, 50, 100

- * 在项目活动中显示的天数

在这里可以设置每页显示多少天之内的活动 默认值:30 表示显示30天之内的活动

- * 主机名称

在这里设置Redmine服务的主机名称和路径。 Redmine用这个值生成邮件签名里的URL, 从而收件人通过点击邮件签名里的URL访问系统。

- * 协议

协议指邮件通知里生成的链接打开方式。 默认值:http

- * 文本格式

格式文本用于问题, 新闻, 文档, Wiki等等的描述字段。

默认只有 textile 一种文本过滤器。

- * 压缩Wiki历史文档

对Wiki历史进行压缩(减小数据库的存储大小)。 默认值:禁用

- * RSS Feed内容条数限制

RSS feeds内容条数的最大值。 默认值:15

- * 允许直接显示的最大文本文件

限制可直接显示的文本文件的最大尺寸 默认值:512KB

- * 查看差别页面上显示的最大行数

限制可最多显示差异行数的最大值, 多于最大行数的内容将被Redmine自动截断 默认值:1500

- * 显示

- * 主题

可供选择的主题样式

Redmine除默认主题外还提供了另外两个备用主题:

- * alternate

该主题的最大特点是根据问题的优先级在问题列表页面为问题着色。

- * classic

该主题追溯到Redmine 0.5.1, 给人一种经典的感觉

- * 默认语言

当应用程序不能识别用户浏览器的语言时, 默认语言将被作为客户端语言。 当发送邮件时也会根据默认语言作为给用户发送邮件的语种。

默认值:en

- * 日历开始于

设置星期的开始日期, 提供的选项有:

- * 根据用户的语言

- * 星期一

- * 星期天

默认值:根据用户的语言

- * 日期格式

设置日期显示格式:

- * 根据用户的语言

根据用户语言定义的格式显示

- * 其他格式

默认值:根据用户的语言

- * 时间格式

设置时间显示格式:

- * 根据用户的语言

- * 其他格式

默认值:根据用户的语言

- * 用户显示格式

设置用户名的显示格式

- * 使用Gravatar用户头像

如果开启, 在应用程序的一些地方可以看到用户的Gravatar头像

- * 默认的Gravatar头像

设置那些没有Gravatar图片的用户显示的图片

- * 认证

- * 要求认证

如果选中, 匿名用户将不能访问系统, 用户必须登录后才能访问。

- * 自动登录

- * 允许自注册

设置是否允许用户注册:

- * 禁用

不允许用户注册

- * 通过邮件认证激活账号

用户注册后将收到一封激活账号的邮件, 里面含有一个激活账号的链接, 点击那个链接就能激活刚注册的账号。

- * 手动激活账号 默认值

用户注册后需要经管理员审核批准。

用户注册后, 管理员会收到一封激活账号的邮件。

- * 自动激活账号

用户注册成功后, 直接就能用刚注册的账号登录系统。

- * 最短密码长度

设置密码的最短长度 默认值:4

- * 忘记密码

如果选中, 在登录界面上将显示一个“忘记密码”的链接。 默认值:开启

- * 允许使用OpenID登录和注册

设置是否允许使用OpenID登录和注册

Note

使用该功能需要安装ruby-openid包，一条简单的gem 安装命令就能搞定(安装后需要重启服务生效)：

```
gem install ruby-openid
```

- * 启用REST web service

设置是否启用REST web service

在和外部系统集时才需要开启该选项

- * 项目

- * 新建项目默认为公开项目

默认情况下，新建的项目为公开项目

你可以在创建项目时不选中该选项，从而创建非公开项目，也可以修改已创建的公开项目为非公开项目。

- * 新建项目默认启用的模块

设置新建项目默认开启的模块

- * 顺序产生项目标识

选中之后，Redmine将为你提供连续的项目标识符。

该标识符用于Redmine REST架构的URL标识用

- * 非管理员用户新建项目时将被赋予的（在该项目中的）角色

该选项用于指定非管理用户在新建项目中的角色。

如果你不指定，系统自动把该用户指定为新建项目的 管理员角色 。

Note

前提是你必须在 [权限](#) 设置界面为非管理员用户赋予创建项目的权限

- * 问题跟踪

- * 允许不同项目之间的问题关联

如果选中，那么不同项目之间的问题可以建立关联。 默认值:禁用

- * 在项目页面上默认显示子项目的问题

如果选中，在浏览父项目的问题列表表示，日历和甘特图时子项目的问题也显示出来。 默认值:开启

- * 计算问题完成度

设置计算问题完成度的方案：

- * 使用问题(的完成度)属性

- * 使用问题的状态

- * 问题输出条目的限制

设置导出问题条目的最大值(包括CSV和PDF)。 默认值:500

- * 问题列表中显示的默认列

设置默认情况下，问题的哪些列被显示。

用户自定义的属性只有标记为 [用于所有项目](#) 才会在这里列出供选择

- * 邮件通知

- * 邮件发件人地址

设置Redmine发送邮件的发件人地址

- * 使用密件抄送 (bcc)

密件抄送即邮件头部隐藏接收者的地址

- * 纯文本 (无HTML)

如果选中, 则邮件仅发送纯文本格式, 不发送HTML格式

- * 选择需要发送邮件通知的动作

设置Redmine的哪些活动事件会触发邮件通知

- * 邮件签名

如何配置发送邮件, 参见 [配置邮件发送](#)

- * 接收邮件

- * 在这些行之后截断邮件

这些设置可以删除接收邮件的签名部分

- * 启用用于接收邮件的服务

从 0.8 之后, Redmine支持通过发送邮件创建问题和评论。

想要使用此功能, 你需要开启接收邮件的API。 默认值: 禁用

- * API key

点击“生成一个key”链接, 系统自动生成一个key, 这是Redmine通过接收邮件创建问题的密钥。

如何配置通过接收邮件提交问题, 参见 Redmine [接收邮件配置](#)

- * 版本库

- * 自动获取程序变更

如果选中, 当用户浏览版本库时应用程序会自动抓取版本库新的变更 默认值: 开启

管理员可以关闭此选项, 通过使用cron在后台定期执行 `Repository#fetch_changesets` 命令抓取版本库新的变更。 例如:

```
ruby script/runner "Repository.fetch_changesets" -e production
```

你也可以通过版本库的 `post-commit` 或者 `post-receive` 钩子调用这个任务, 从而使每次提交之后, 变更集自动被抓取。

- * 启用用于版本库管理的Web Service

该选项只有在你安装了自动创建SVN版本库的脚本时开启。 默认值: 禁用

- * 启用 SCM

设置开启的SCM类型

- * 版本库编码

设置版本库里文件的编码方式 (多个值之间用逗号分开)。 这些编码用于将文件的内容和diff转换成UTF-8, 从而确保浏览版本库时正确显示。

当填写多个编码方式时, 第一个编码是文件使用的编码

- * 提交注释的编码

- * 在文件变更记录页面上显示的最大修订版本数量 默认值: 100

- * 在提交信息中引用和解决问题

当版本库抓取变更集时, 会扫描提交信息中引用和解决的问题ID。

* 在这里管理员可以定义提交注释中引用和解决问题的关键字, 从而通过使用这些关键字自动使该次修订引用或者解决某些问题。

同样可以设置使用解决问题的关键字后问题状态的变更, 以及完成度的变更 * 默认的关键字有:

* 引用: refs, references, IssueID

* 解决: fixes, closes

* 用于解决问题的关键字后问题的变更设置:

* 应用后的状态

* 完成度

* 示例

This commit refs #1, #2 and fixes #3

说明:

* 这次代码提交引用了问题 1, 2 同时解决了问题 3

* 每个关键字和问题ID之间可以用空格, 逗号或者&分开

10 高级配置

10.1 版本库配置

* 常规信息

Redmine一直以来支持不同的版本控制工具集成: Subversion, CVS, Mercurial, Darcs, Git和Bazaar。

首先你要在Redmine主机上安装合适的二进制包, 并确保这些二进制文件在PATH环境变量中可用:

SCM	已测试过的版本	说明
Subversion	1.3 & 1.4 & 1.5 & 1.6	需要1.3或者更高的版本
CVS	1.12.12	需要1.12, 不能同时使用CVSNT
Mercurial	0.9.3	
Bazaar	1.0.0.candidate.1 & 2.0.4	
Darcs	1.0.7	
Git	1.5.4.2	

比如, 如果你想用Redmine访问Subversion版本库, 你必须在Redmine主机上安装SVN二进制文件。

* 给项目添加一个已经存在的版本库

在项目配置选项卡下有 版本库 子选项库, 在这里可以为项目添加版本库。

选择你所使用的版本库类型, 并输入版本库路径或者URL

Note

当配置完版本库第一次访问时, Redmine将抓取版本库中已经存在的所有提交信息, 并存入数据库。所以如果你的版本库特别大, 那么该过程将会很长。

为了避免这种情况, 你可以将该过程放在后台执行。

在配置完版本库后, 运行下面的命令:

```
$ ruby script/runner "Repository.fetch_changesets" -e production
```

所有的提交将会被存入数据库

从Redmine0.9之后, 你还可以通过一个链接给某一项目或者所有项目执行 fetch_changesets:

```
http://redmine.example.com/sys/fetch_changesets (给所有活动的项目取回变更集)
http://redmine.example.com/sys/fetch_changesets?id=foo (仅给foo项目取回变更集)
```

* Subversion 版本库

Redmine支持的协议有([http:](http://), [svn:](svn://), [file:](file://)), 只需要输入版本库的URL

例如:

```
http://host/path/to/the/repository
```

如果版本库需要身份认证, 你需要额外输入用户名和密码

Note

如果你想要用 `svn+ssh://` 访问版本库, 你必须把`svn+ssh`配置为非交互的方式。

同时需要设置ssh认证的 `public/private` 钥对

Hint

Subversion 基于路径的授权

Redmine对Subversion授权是这样处理的: 在配置Subversion版本库时, 除了输入URL外, 还需填写一个拥有Subversion读权限的账户信息(如果你的Subversion不允许匿名访问)。显然很多商业公司的代码有很高的严密性, 大多是不允许匿名访问的, 看来是必填了。而Redmine对Subversion认证不严格也是从这里开始的。填写了这个账户之后, 不管哪个用户登录Redmine浏览版本库, 看到的版本库信息都是该账户应该能看到的, 而不是当前登录用户应该看到的。也就是说, 在这种情况下, 如果你想对代码浏览严加控制, 那么就需要创建一个很低权限的Subversion账号, 用这个账号作为Redmine访问Subversion的账号。但这样一来, 那些具有高权限的用户就不乐意了(他们本来是能看到很多代码的, 但这样一来他们的权限无形之中被剥夺了)。

为了满足商业软件公司的需求, 群英汇对Redmine关于Subversion授权的部分进行了扩展, 采用Subversion本身的授权模式。我们添加了解析Subversion授权文件的组件, 可以很好地解析每个用户的权限, 从而保证Redmine浏览权限和Subversion定义的权限一致, 完美解决了Redmine中Subversion授权不严格的问题, 实现了Subversion基于路径的授权。

配置步骤, 参见 [Subversion基于路径授权](#)

* CVS 版本库

输入:

* 版本库的URL (也可以输入路径或者链接字符串, 例如 `:pserver`)

* 模组名称

示例:

```
:pserver:login:password@host:/path/to/the/repository
```

以下几种版本库输入版本库的路径即可

* Git 版本库

* Mercurial 版本库

Hint

解决了中文版Mercurial版本库无法使用的问题

由于Redmine在匹配Mercurial版本信息时是基于英语匹配的, 所以当我们使用中文Mercurial时, 它会抛出匹配失败的异常, 造成中文用户不能使用。

* Bazaar 版本库

* Darcs 版本库

10.2 发送邮件配置

首先将 `config/email.yml.example` 重命名为 `config/email.yml`

然后配置发送邮件需要的参数：

```
production:
  delivery_method: :smtp
  smtp_settings:
    address: mail.foo.bar
    port: 25
    domain: foo.bar
  # authentication: :login
  # user_name: xxx
  # password: xxx
```

注意：

用#注销掉的那三行信息，是邮件服务器身份验证信息。如果你的用户已经使用了域验证，就不需要那三行信息。

访问redmine，以管理员的身份登录系统，点击“管理”——>“配置”——>“邮件通知”，然后点击右下角的“发送测试邮件”，就可以测试你的邮件服务配置是否成功了。

Note

当`delivery_method`设置为`smtp`时，它的处理流程是这样的：在触发邮件发生的事件发生后，首先发送邮件，直到邮件发送完毕，页面才跳转。这就是说在发送邮件的过程中，页面一直处于等待状态。那么有没有一种异步发送邮件机制呢？在Redmine 0.9 之后，可通过将`delivery_method`设置为`async_smtp`，这时发送邮件和页面跳转属于不同的线程，当触发邮件的事件发生后，发送邮件将会通过消息队列在后台执行，而页面继续向下执行，不用等到邮件发送完毕。

10.3 接收邮件

从Redmine-0.8.0 开始，通过配置，你可以以发送邮件的方式创建或者评论问题(issue)。

10.3.1 方案

你可以通过以下几种方式配置Redmine接收邮件的功能：

- * 从邮件服务器转发邮件
- * 优点：与远程邮件服务器协作，邮件处理即时，快速(无环境重载)
- * 缺点：需要一些邮件传输代理(MTA)的配置(如Postfix, Sendmail...)
- * 从IMAP或者POP3服务器读取邮件
- * 优点：易于安装，不需要配置邮件传输代理(MTA)，与远程邮件服务器协作
- * 缺点：邮件处理不即时(需要添加一个定期读取邮件的 `con job`)
- * 从标准输入读取邮件
- * 优点：用于测试的最佳选项
- * 缺点：慢(每次读取邮件时都要重新载入环境)，需要配置邮件传输代理(MTA)

10.3.2 各种方案的详细说明

* 从邮件服务器转发邮件

用一个独立的脚

本转发邮件服务器发来的邮件。这个脚本从标准输入读取原始邮件并通过HTTP请求转发给Redmine。该脚本在redmine目录下：`extra/mail_handler/rdm-mailhandler.rb`

使用这项功能之前，你需要开启接收邮件的API：

开启操作：“管理”->“配置”->“接收邮件”，选中开启接收邮件服务的复选框，并点击链接“生成一个key”，产生一个密钥。

复制`rdm-mailhandler.rb`到你的邮件服务器，确保该脚本有执行的权限并配置你的邮件传输代理(MTA)。

用法：

```
rdm-mailhandler [options] --url=<Redmine URL> --key=<API key>
必须选项：
-u, --url                Redmine服务的URL
-k, --key                Redmine API key
常用可选项：
-h, --help              显示帮助信息
-v, --verbose           显示额外的信息
-V, --version           显示版本信息
```

问题属性可控的选项：

```
-p, --project=PROJECT    项目的标识符
-t, --tracker=TRACKER   跟踪标签的类型
--category=CATEGORY     问题类别的名称
--priority=PRIORITY     优先级的名称
-o, --allow-override=ATTRS 允许在邮件里覆盖先前已指定的属性，多个属性之间用逗号分开
```

示例：

```
# 没有指定项目，那么在邮件内容里必须包含“Project”关键字，否则Redmine将不处理该邮件
rdm-mailhandler --url http://redmine.domain.foo --key secret
# 设置默认的项目和跟踪标签类型，同时可以在邮件内容里覆盖tracker和priority选项
rdm-mailhandler --url https://domain.foo/redmine --key secret \
  --project foo \
  --tracker 错误 \
  --allow-override tracker,priority
```

下面是一个Postfix的例子：

```
foo: "|/path/to/rdm-mailhandler.rb --url http://redmine.domain --key secret --project foo"
```

* 从IMAP服务器读取邮件

redmine提供了一个rake任务(`redmine:email:receive_imap`)负责从IMAP服务器读取邮件。

当cron 运行该rake命令时你需要使用选项 `-f /path/to/redmine/appdir/Rakefile`，因为不这样的话会出现找不到rakefile的异常。

下面是一个定期每30分钟读取一次邮件的cron文件：

```
*/30 * * * * redmineuser rake -f /path/to/redmine/appdir/Rakefile redmine:email:receive_imap \
RAILS_ENV="production" host=imap.foo.bar username=redmine@somenet.foo password=xxx
```

在cron文件中，这个命令必须位于一行上

如果用Windows作为服务器，`pycron`可以用于调度读取邮件的任务。

同时需要你修改防火墙设置，打开同通过TCP链接IMAP的143端口。

可用的IMAP选项：

```
host=HOST                IMAP服务主机(*默认值: 127.0.0.1*)
port=PORT                IMAP服务端口(*默认值: 143*)
ssl=SSL                  是否使用SSL? (*默认值: false*)
username=USERNAME       IMAP账号
password=PASSWORD       IMAP密码
folder=FOLDER           将要读取的IMAP文件夹(*默认值: INBOX*)
move_on_success=MAILBOX 将已成功读取的邮件移动到MAILBOX中，而不是删除掉
move_on_failure=MAILBOX 将Redmine忽略的邮件移动到MAILBOX中
```

问题(issue)属性可控选项:

project=PROJECT	项目标识符
tracker=TRACKER	跟踪标签类型
category=CATEGORY	问题类别名称
priority=PRIORITY	优先级名称
allow_override=ATTRS	允许在邮件内容里覆盖先前已指定的属性, 多个属性之间用逗号隔开

rake命令的一个例子:

```
# 没有指定项目, 这时邮件内容里必须包含“Project”关键字
rake redmine:email:receive_imap RAILS_ENV="production" \\  
  host=imap.foo.bar username=redmine@somenet.foo password=xxx
# 指定默认的项目和跟踪标签类型, 同时允许在邮件里覆盖tracker和priority属性的值
# both tracker and priority attributes:
rake redmine:email:receive_imap RAILS_ENV="production" \\  
  host=imap.foo.bar username=redmine@somenet.foo password=xxx ssl=1 \\  
  project=foo \\  
  tracker=错误 \\  
  allow_override=tracker,priority
# 将成功处理的邮件转移到“read”mailbox中, 失败的则转移到“failed”mailbox中
rake redmine:email:receive_imap RAILS_ENV="production" \\  
  host=imap.foo.bar username=redmine@somenet.foo password=xxx \\  
  move_on_success=read move_on_failure=failed
```

忽略的

邮件(包括未知用户、未知项目以及来自redmine禁用帐户的邮件)被标记为‘已读’,而不是从IMAP服务器中删除。

选项 allow_override 不仅仅覆盖rake的默认值,而且可以覆盖邮件里的所有属性。例如,你要覆盖tracker,你可以添加参数: allow_override=tracker

* 从POP3服务器读取邮件

仅在trunk和将来的1.0版本可用

redmine提供的rake任务(redmine:email:receive_pop3)可以用于从POP3读取邮件

POP3可用的选项:

host=HOST	POP3服务主机(*默认值: 127.0.0.1*)
port=PORT	POP3服务端口(*默认值: 110*)
username=USERNAME	POP3账号
password=PASSWORD	POP3密码
apop=1	是否使用APOP认证(默认值: false)
delete_unprocessed=1	从服务器上删除不能成功处理的消息(*默认情况是继续保存在服务器上*)

关于问题属性的可控选项可以参考上面IMAP章节

* 从标准输入读取邮件

redmine提供的rake任务(redmine:email:receive)用于从标准输入读取单个的原始邮件

问题属性可控的选项:

project=PROJECT	项目标识符
tracker=TRACKER	跟踪标签类型
category=CATEGORY	问题类别名称
priority=PRIORITY	优先级名称
allow_override=ATTRS	允许在邮件内容里覆盖先前已指定的属性, 多个属性之间用逗号隔开

示例:

```
# 没有指定项目, 这时邮件里必须包含“Project”关键
rake redmine:email:read RAILS_ENV="production" < raw_email
# 指定默认的项目和跟踪标签类型, 并且允许在邮件里覆盖tracker和priority选项
# both tracker and priority attributes:
rake redmine:email:read RAILS_ENV="production" \\  
  project=foo \\  
  tracker=错误 \\  
  allow_override=tracker,priority < raw_email
```

选项 allow_override 不仅仅覆盖rake的默认值,而且可以覆盖邮件里的所有属性。例如,你要覆盖tracker,你可以添加参数: allow_override=tracker。

10.3.3 工作原理

收到邮件后, Redmine从发件人地址中查找对应的账号。未知用户和已锁定的用户将被忽略。

如果邮件主题包含像这样的内容“ Re: [xxxxxx #123]”, 那么该邮件将被作为问题(issue)#123的回复处理。

否则将会以邮件主题为问题的标题, 新建一个问题(issue)

* 目标项目:

目标项目可以通过接收邮件里的project选项指定。project应该用项目的ID而不是项目的名称来指定。

如果不使用该选项, 你需要在邮件内容中指定问题关联的项目。你可以在邮件里添加这样一行: “project:foo”。

例子(邮件内容):

```
Project: foo
在这里可以写一些关于问题的描述信息
```

你可以给项目选项指定一个默认的项目并允许并允许用户通过 allow_override 选项覆盖默认的项目。

例子:

```
# 指定默认项目, 并允许用户在邮件里覆盖project属性
rake redmine:email:receive_imap [...] project=foo allow_override=project
```

当然, 用户的权限也会别检查, 那些没有创建问题(issue)的用户发送的邮件将被忽略。

同时必须确保目标项目没有使用没有设定默认值的用户自定义问题(issue)字段, 否则创建问题(issue)将会失败。

* 问题(issue)属性

根据你使用的接收邮件选项(参考 allow_override 选项), 用户可以在提交问题(issue)时覆盖一些属性。

可以在邮件里用以下关键字实现: Tracker, Category, Priority, Status。

例子:

```
# 这是一个新问题, 并将覆盖一些属性
Project: foo
Tracker: 错误
Priority: 紧急
Status: 反馈
```

* 跟踪者

如果发送邮件的用户有“添加问题(issue)跟踪者”的权限, 邮件中To或者Cc字段里的用户将作为新建问题(issue)的跟踪者

* 邮件格式以及附件

Redmine用邮件的纯文本部分填充问题(issues)的描述, 因此当收到一个仅有HTML的邮件, 那些HTML标签将被删除。

邮件的附件将会自动附加给问题(issue), 除非超过了应用程序设置的附件最大值。

10.3.4 例子演示

以从IMAP服务器读取邮件为例

添加一个Redmine从IMAP服务器读取邮件的corn任务

```
wangsheng@pc01:/etc/cron.d$ cat receive_imap
# m h dom mon dow user  command
*/10 * * * * root rake -f /opt/redmine/web/Rakefile redmine:email:receive_imap RAILS_ENV="production" \
host=imap.foo.bar username=xxx@foo.bar password=xxxxxx allow_override=tracker
```

说明:

* 这个任务会每隔10分钟执行一下那个rake命令

* `username`最好跟“管理”->“配置”->“邮件通知”界面里的 邮件发件人地址 相同, 否则当你直接回复邮件更改问题时可能会出现收件人地址不可达的异常(如果你在该界面配置了一个不存在的邮件账号)。最理想的做法是给Redmine创建一个IMAP账号, 写在这里。

好了, 下面发送一个测试邮件:

收件人: rake里的`username`值

邮件标题:

```
# 这将是新建问题的标题
# 这是通过邮件提交的问题
```

邮件内容:

```
# 由于在rake命令里没有指定默认项目, 所以这里必须指定项目
Project: foo
Tracker: 支持
下面可以写一些关于问题的描述
```

说明: 该邮件将会创建一个“支持”类型的问题(issue)

如果10分钟内, 你访问Redmine, 点击问题选项卡, 看到了这个问题, 那么证明已经成功配置。

10.4 发送提醒邮件


Redmine提供了一个rake任务, 它可以发送那些未来几天之内需要完成的问题的提醒邮件。

可用选项:

- * `days`: 设置几日之内的提醒邮件 默认值:7
- * `tracker` 跟踪标签的ID 默认:所有跟踪标签
- * `project`: ID或者项目标识符 默认:所有项目

下面是一个发送未来7天之内需要解决的问题的提醒邮件示例, 问题的指派者将收到这封邮件:

```
rake redmine:send reminders days=7 RAILS_ENV="production"
```

发件人 redmine@bj.ossxp.com 
标题 5 个问题需要尽快解决

指派给您的 5 个问题需要在 7 天内完成:

- gitProject - [错误 #3](#): 测试邮件1
- gitProject - [错误 #5](#): 邮件测试3
- svnProject - [错误 #18](#): task1
- svnProject - [错误 #19](#): task2
- svnProject - [错误 #21](#): task4

[查看所有问题](#)

您收到这个通知是因为您或者订阅, 或者参与到流程中了。要修改您的通知相关设置, 访问: <http://hostname/my/account>

提醒邮件截图.

10.5 LDAP 认证

Redmine一直以来都支持一个或者多个LDAP认证源。

以管理员身份登录系统, 点击 管理 -> 配置 -> 认证 -> LDAP 认证, 即可进入LDAP认证页面。

* 新建LDAP认证

在LDAP认证页面, 点击右上角的 新建认证模式 链接, 进入新建LDAP认证的页面:

* 名称

任意给你新建的LDAP起个名称

* 主机

LDAP主机名称, 也可以写成LDAP主机的IP

* 端口

LDAP端口, 默认是 389

* LDAPS

如果你想用LDAPS访问LDAP目录, 那么你需要选中它

* 账号

如果LDAP服务允许匿名用户访问, 则不填; 否则需要填写一个能访问LDAP的用户名

* 密码

和上面的账号一起设置

* Base DN

LDAP目录树的顶级DN

* 即时用户生成

选中该选项后, 任何LDAP用户第一次登录Redmine后, Redmine会将LDAP里“登录名, 名字, 姓氏, 邮箱地址”这些字段复制到Redmine的数据库中, 从而为这些用户创建了Redmine账号。

* 属性区域

在这里分别填写LDAP的登录属性

* 登录名

* 名字

* 姓氏

* 邮件地址

一个典型的例子:

```
Name      = My Directory
Host      = host.domain.org
Port      = 389
LDAPS     = no
Account   = MyDomain\UserName (or UserName@MyDomain depending on AD server)
Password  = <password>
Base DN   = CN=users,DC=host,DC=domain,DC=org

On-the-fly user creation = yes
Attributes
Login     = sAMAccountName
Firstname = givenName
Lastname  = sN
Email     = mail
```

创建之后页面自动跳转到LDAP认证页面, 点击“测试”链接测试你的设置是否能够和LDAP服务器链接上

Note

* Base DN 变种

如果链接失败，你可以将Base DN字段改为“DC=host, DC=domian, DC=org”

* LDAP属性名称大小写敏感

认证模式 (LDAP)

新建LDAP截图

10.6 配置单点登录

前提

* 已经配置好LDAP 参见 [LDAP认证](#)

* 安装配置Cosign

单点登录配置步骤:

以管理员身份登录Redmine，点击“管理”-->“配置”-->“认证”，进入认证配置页面
在页面的下方区域就是配置单点登录的地方：

配置

配置单点登录截图

* 单点登录模式

选择是否开启单点登录。如果开启，你需要根据安装Cosign的具体版本选择合适的Cosign登录模式。

* 禁用 默认值 * CoSign 3.x 登录模式 * CoSign 2.x 登录模式

- * 登录URL
- * 退出URL
- * 单点登录服务名

10.7 从LDAP同步账号

前提:

已经配置好LDAP 参见 [LDAP认证](#)

操作步骤:

以管理员身份登录Redmine, 点击“管理”-->“配置”-->“认证”-->“LDAP认证(页面右下角)”, 进入LDAP认证页面。



从LDAP同步账号截图

如果你已经配置有LDAP认证, 你会发现在LDAP认证列表中有一个“从LDAP添加用户”的链接, 点击它, 将跳转到 从LDAP添加用户 的页面:

* 上面是一个过滤器

在这里可以对用户列表进行过滤。比如你想查找用户 zhangsan 可以这么做:

在过滤器的输入框中输入 uid=zhangsan, 点击“应用”按钮, 这时用户列表就仅有 zhangsan 一个人的用户信息了。

* 下面是LDAP用户信息列表

想要同步账号, 只需点击前面那个复选框, 然后点击“同步账号”按钮, 系统将自动完成Redmine账号的建立。

10.8 Subversion 基于路径授权

配置过程:

以管理员身份登录Redmine, 选择用Subversion作版本控制的项目, 点击“配置”-->“版本库”, 进入版本库配置页面。

选择Subversion, 这时你会发现多了两个属性:

* 授权文件

在这里输入Subversion授权文件 (svn.access) 的全路径。

* 授权模组名称

输入你希望通过Redmine浏览的项目 (Subversion中每一个模组即代表一个项目)。



The screenshot shows the '配置' (Configuration) page in Redmine. At the top, there is a navigation bar with links: 概述, 活动, 路线图, 问题, 新建问题, 新闻, 文档, Wiki, 文件, 配置. Below this, there is a sub-navigation bar with links: 信息, 模块, 成员, 版本, 问题类别, Wiki, 版本库, 讨论区, 活动 (时间跟踪). The main configuration area is titled '配置' and contains the following fields:

- SCM: Subversion (dropdown menu)
- URL: (with a hint: (file:///, http://, https://, svn://, svn+[tunnelscheme]://))
- 登录名:
- 密码:
- 授权文件:
- 授权模组名称:

At the bottom left of the configuration area, there is a '创建' (Create) button.

配置Subversion基于路径授权截图

Note

如果你用的

是http协议，而且svn不允许匿名访问，那么为了让所有用户都能有效地看到自己应该看到的版本库信息，你需要创建一个对整个版本库读权限的账号，输入到登录名和密码部分。因为我们的Subversion权限检查组件是在Redmine原来的基础上，对输出的信息再次进行过滤。

所以如果你在这里输入的账号权限太低，

还是会出现一些高权限的用户无法看到他们应该看到的内容的现象。